

## DATA CLEANING

### Task-1

--->Data Cleaning for 'agedistribution\_2016\_estimates' is a process of renaming the appropriate columns in a dataset containing data. This improves the quality and usability of the data for analysis and modeling purposes

```
import pandas as pd

import warnings

warnings.filterwarnings("ignore")

def read_data_from_csv():

    #Read the dataset

    df = pd.read_csv('agedistribution_2016_estimates.csv')

    return df

def rename_columns():

    data=read_data_from_csv()

    data.rename(columns={'Age': 'Age_group','M': 'Male','F': 'Female','Tot': 'Total'},inplace=True)

    return data

def task_runner():

    rename_columns()
```

### Task-2

--->Data cleaning for ""death\_and\_recovery"" involves removing unwanted columns from a dataset. For the purposes of analysis and modelling, this enhances the quality and usability of the data.

```
import pandas as pd

import warnings

warnings.filterwarnings("ignore")

#Loading the data
```

```

def read_data_from_csv():
    df = pd.read_csv('death_and_recovery.csv')
    return df

def data_cleaning():
    data = read_data_from_csv()
    data.drop(columns=['comorbidity','State_code'],axis=1,inplace=True)
    return data

def task_runner():
    data_cleaning()

```

### Task-3

--->Data Cleaning for 'hospitalbeds' is a process of renaming the appropriate columns in a dataset containing data. This improves the quality and usability of the data for analysis and modeling purposes.

```

import pandas as pd
import warnings
warnings.filterwarnings("ignore")

def read_data_from_csv():
    #read the dataset
    df = pd.read_csv('hospitalbeds.csv')
    return df

def rename_columns():
    data=read_data_from_csv()
    data.rename(columns={'serial': 'sno','state': 'State_UT','Hosp_Aval': 'Hospitals_Available',
    'Beds_Aval': 'Beds_Available','Pop_beds':'Population_beds'},inplace=True)
    return data

def task_runner():
    rename_columns()

```

#### Task-4

--->Data cleaning for ""statewisedata"" involves removing unwanted columns from a dataset. For the purposes of analysis and modelling, this enhances the quality and usability of the data.

```
import pandas as pd

import warnings

warnings.filterwarnings("ignore")

def read_data_from_csv():

    df = pd.read_csv('statewisedata.csv')

    return df

def data_cleaning():

    # DO NOT REMOVE FOLLOWING LINE

    data = read_data_from_csv()

    data.drop(columns=['Delta_Confirmed','Delta_Recovered','Delta_Deaths'],axis=1,inplace=True)

    return data

#Do not Delete the Following function

def task_runner():

    data_cleaning()
```

#### Task-5

--->Utilize the MySQL database information provided in ""Database info"" to manually create the following tables for the cleaned dataset

```
agedistribution_2016_estimates

death_and_recovery

hospitalbeds

datewisepatients

icmrtestingdata

statewisedata
```

## MINE INFORMATIONAL INSIGHTS USING SQL

### Task-1

--->Display the states, gender affected and the confirmed cases in their respective states where confirmed cases are more than 100.

```
select Distinct(d.State), d.Gender, s.Confirmed from death_and_recovery as d INNER JOIN statewisedata as s on s.state_UT=d.state where s.Confirmed>100
```

### Task-2

--->Which states have collected more than 1000 samples in a day? Provide the serial number, state name, and the total number of samples tested for each state, using data from the 'icmrtestingdata' and 'statewisedata' tables.

```
select s.sno, s.State_UT, i.TotalSamplesTested from icmrtestingdata i INNER JOIN statewisedata s on s.sno=i.sno where i.TotalSamplesTested>1000
```

### Task-3

--->Display the patient status in each state from the death\_and\_recovery table

```
SELECT t1.Patient_status, t2.City, t1.Age FROM death_and_recovery AS t1, death_and_recovery AS t2 WHERE t1.State = t2.State
```

### Task-4

--->Display the hospital beds along with their location where patients have recovered from covid-19 and those beds are made available to the needy patients waiting in the queue to get admitted.

```
select d.Patient_Status, d.State,d.City,h.Beds_Available from death_and_recovery as d inner join hospitalbeds as h on d.State=h.state_Ut where d.Patient_Status="recovered"
```

### Task-5

--->Display the total number of people in assam who have recovered

```
select COUNT(Patient_status) from death_and_recovery where patient_status='recovered' and state='Assam'
```

#### Task-6

--->Show the state, hospitals and beds available where population beds and hospitals available are more than 1000.

```
select State_UT, Hospitals_Available, Beds_Available from hospitalbeds where Hospitals_Available>=1000 and Beds_Available>=1000 and State_UT<>'india' and State_UT<>'kerala'
```

#### Task-7

--->Show states where active cases are less than 50

```
select state_UT from statewisedata where active<50
```

#### Task-8

--->Which dates are associated with the availability of beds, as captured in the 'datewisepatients' and 'hospitalbeds' tables?

```
select d.date,h.beds_available from hospitalbeds h , datewisepatients d group by d.date,h.beds_available
```

#### Task-9

--->Show the details of the number of samples tested across each timestamp

```
select UpdatedTimeStamp ,TotalSamplesTeste from icmrtestingdata where TotalSamplesTested>0
```

#### Task-10

--->Display the number of males and females who have recovered

```
select Gender,count(*) from death_and_recovery where Patient_Status='recovered' Group by 1
```

#### Task-11

--->List the states where the population is greater than the number of beds available in descending order of serial number

```
select State_UT,Beds_Available from hospitalbeds where Beds_Available<population_beds order by sno desc
```

#### Task-12

--->What is the total number of samples tested, total number of positive cases, and the difference between the total samples tested and total positive cases in the 'icmrtestingdata' table?

```
select TotalSamplesTested, TotalPositiveCases, (TotalSamplesTested-TotalPositiveCases) from icmrtestingdata
```

#### Task-13

--->Find the number of hospital beds available in each state

```
select Beds_Available,State_UT from hospitalbeds
```

#### Task-14

--->Display the total number of beds available in Tamil Nadu

```
select Beds_Available from hospitalbeds where State_UT="tamil nadu"
```

#### Task-15

--->Display the total number of beds available in India.

```
select sum(Beds_Available) from hospitalbeds
```

#### Task-16

--->What are the distinct values of 'TotalSamplesTested', 'TotalPositiveCases', and 'UpdatedTimeStamp' in the 'icmrtestingdata' table?

```
select TotalSamplesTested, TotalPositiveCases, UpdatedTimeStamp from icmrtestingdata group by  
TotalSamplesTested, TotalPositiveCases, UpdatedTimeStamp
```

#### Task-17

--->Display the total confirmed cases till 31-March in Maharashtra

```
select SUM(active_Cases) from icmrtestingdata where state=Maharashtra
```

#### Task-18

--->Calculate the summing distribution of males and females aged 0 to 49 who have been impacted by COVID-19.

```
select sum( Male), sum(Female) from agedistribution_2016_estimates where Age_group <'50-54'
```

#### Task-19

--->Find out the recovery rate among the states and display it along with the names of the states and the number of recovered & active cases.

```
select Recovered, Active, State_UT,(Recovered/Active)*100 from statewisedata
```

#### Task-20

--->Display the states along with the ratio of Beds available against the total population beds

```
select State_UT, Beds_Available, Population_beds, Beds_Available/Population_beds from hospitalbeds
```

#### Task-21

--->What are the different patient statuses and the corresponding cities recorded in the 'death\_and\_recovery' table, after joining it with the 'statewisedata' table based on the matching State\_UT values?

```
select distinct(death_and_recovery.Patient_status), death_and_recovery.City from death_and_recovery inner join statewisedata on death_and_recovery.state=statewisedata.State_UT
```